Implementation of Load Balancing Technology Using Raspberry Pi as a Server for Computer Based Examination

Budi Tjahjono, Ade Sulaeman and Fransiskus Adikara
Faculty of Computer Science, Esa Unggul University, Jl. Terusan Arjuna, Jakarta, Indonesia
{budi.tjahjono, ade.sulaeman, fransiskus.adikara}@esaunggul.ac.id

Keywords: load balancing technology, raspberry pi, computer networking.

Abstract: Since 2014, Indonesian government has implemented a computer-based National Examination as a substitute for paper-based National Examination. Schools that conduct computer-based National Examinations should provide the Server as a provider of questions for students to work on. Not all schools have sufficient funds for server procurement. This research demonstrates load balancing technology in the server of computer-based examination at an educational institution using Raspberry Pi as a first step to handling the computer-based National Examination. Raspberry PI is a relatively low-cost single-board computer technology that will serve as a solution to reduce costs for conductiong the computer-based examination. This research using PPDIOO method as research methodology to design and implement the Raspberry Pi as a load balancing server for computer-based examination server. The result show that Rasberry Pi can be use as server and the load balancing technology can be also implemented to improve the access quality of user in the server.

1 INTRODUCTION

Computer-based exams are derivatives that refer to the Computer-Based National Examination which since 2014 has been implemented by the government as a Computer Based Test (CBT) Test and replaces the National Examination (UN) system based on paper. The implementation of UNBK currently uses a semi-online system where the exam is sent from the central server in real time through the network to be synchronized to local servers in schools. The student exam will be served by a local server offline. When finished, the test results are sent back from the local server to the central server online (Ministry of Education and National Culture of the Republic of Indonesia, 2017).

Based on data from the Ministry of Educationa and National Culture (Kemendikbud) of the Republic of Indonesia as of February 5, 2018 stated that in academic year of 2016/2017 there were 55,802 schools that could not implement UNBK. This figure is greater than the number of schools that can implement UNBK, namely 23,342 schools throughout Indonesia. While there are 3,682 schools that join to implement UNBK by referring to schools that have been able to implement UNBK. The data illustrates that there are still many schools in Indonesia that cannot implement UNBK.

This UNBK turned out to motivate several schools to create a system similar to UNBK as a school examination system. By making a similar system as a first step, it is hoped that further UNBK implementation can run better. In addition, the computer-based examination system will facilitate teachers in correcting values (Susanti, 2016).

Computer-based examination systems that have been used in several schools today contain exam questions that are planted on the server and can then be accessed by students through another computer (client). This exam system uses client-server architecture. This architecture places a computer as a server. This server is in charge of providing services to other terminals that are connected in a network system or what we refer to as ‘client’

The use of servers on the local network, such as those used in this computer-based test, can certainly receive a very large traffic load. This is because students exam will access the server at the same time. So that a surge in traffic will be too much to be handled by the web server and causes the server to overload. This will interfere with the system of the exam itself. Therefore it is necessary to have a way to overcome the overload.

A solution that can be done to improve the server's ability to service requests is by upgrading the server's hardware. The solution is still lacking, because a server has hardware limitations that can be installed on one server. Another solution that can be done to meet the request of the client is by adding a new server unit and applying the clustering method, where several servers serve client requests evenly so that the number of current connections can increase and the availability of the server is higher. (B. M. Moniruzzaman, et al., 2015).

One of the type of clustering is called Load Balancing. Load Balancing is a technique for dividing load on multiple servers. Load Balancing is suitable for use on networks with high traffic. With this technology, the server load can be shared and the performance of computer-based exams can be maximized.

In building a server on a local network, a server that is flexible and easy to carry around is needed. This needs to be considered because the computer-based exam process is usually done in class. With this in mind, a server with flexible specifications is needed to adjust to the place.

The development of Raspberry PI as a web server and load balancer is very interesting to be used as research because Raspberry PI as a small computer that has a linux-based operating system, does not require large power and data storage power to be operated into server clusters. (Putra & Sugeng, 2016). With this Raspberry Pi, server on computer-based exams will be built.

The purpose of this study is to make Raspberry Pi as a load balancing server so that it can become a high availability server and reduce the cost of procuring infrastructure to create a computer-based exam system. The next section will discuss the previous research, followed by research methodology and the steps taken to design and use Raspberry Pi as a server. The results of the study will be discussed in the fourth and final section, concluded in conclusion part and subsequent work.

2 RELATED WORK

Rasberry Pi…..

Research related to Load Balance using any kinds of Raspberry Pi ...

3 RESEARCH METHODOLOGY

This server design will be made with Cisco Lifecycle Service or PPDIOO method. The following is a framework of thought which describes the steps of the method for developing the system :

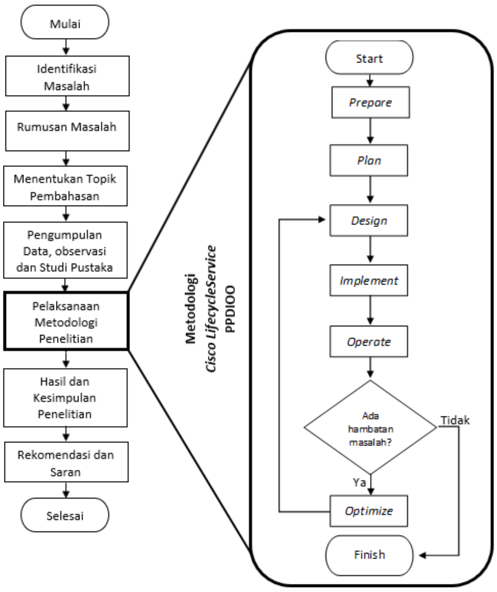

Figure 1: This caption has one line so it is centered.

4 IMPLEMENTATION OF RASPBERRY PI AS SERVER

In this section, detailed steps are carried out according to the research methodology chosen to implement the Raspberry Pi as a server that can be used as a computer-based Exam server.

4.1 Preparation

The initial process is to carry out problem analysis from existing sources and data is the first step taken. This process will analyze the extent of the existing system regarding computer-based exams.

Topology design for Computer Based Exams is almost the same as UNBK, but does not use the Internet (Figure 2).

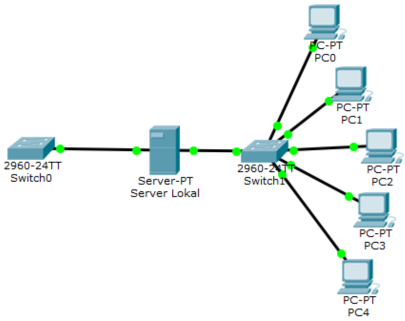

Figure 2. UBK Topology

In addition to topology, the following are specifications of the server and client computers used in computer-based exams as shown in Table 1

Table 1. Hardware Specifications

	Server	Client
Type	Dell Built Up	Standard Desktop
Operating System	Ms. Windows Server 2012	Windows 7
RAM	8 GB	4 GB
Processor	Xeon	Intel Core i3
Price	Rp 10.990.000	-

4.2 Plan

In this study, the step of using computer software is as follows:

a. The operating system used is Raspbian Jessie for servers and Windows 7 as minimum requirement as client.

b. The Web Server used is Nginx, included for load balancing.

c. The web application that will be used is the English Gate web application with MySQL and PHP databases.

d. Google Chrome web browser with Flash Player add ons.

e. PuTTY will be used for terminal access from administrator PC to Raspberry PI.

f. The Webserver Stress Tool will be used as a test application.

g. Some simple scripts that are used to simplify testing.

The specification of the hardware used in this study can be seen in Table 2.

Table 2. List of Hardware

No.	Hardware	Qty	Function	Description
1.	Raspberry PI	4 units	Web Server, Load Balancer, Database Server	Type : Raspberry Pi 3 Model B
2.	Laptop / PC	2 units	Computer administrator and Tester Computer	OS : minimum Windows 7
3.	Switch	2 units	Connecting LAN cable to the computer	-
4.	LAN Cable	-	Connect between devices	Cat 5e
5.	PC Client	30 units	Used for student exams	-

Computer networks in this study were described as follows :

a. The type of computer network used is client-server.

b. The Network Topology used is the Bus Topology on the client network and the Star topology on the server network.

c. The computer cluster technique will be used which will form the server design.

d. The type of cluster computer for the server to be used is load balancing.

4.3 Design

The topology design that will be implemented in a computer laboratory as shown in Figure 3.

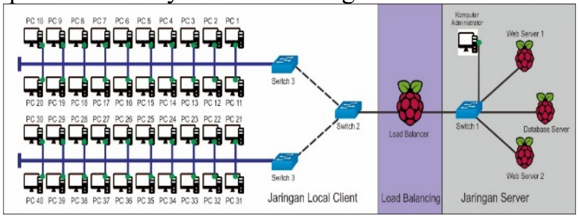

Figure 3. UBK Topology

The diagram block that will be used in load balancing using Raspberry Pi is shown in Figure 4.

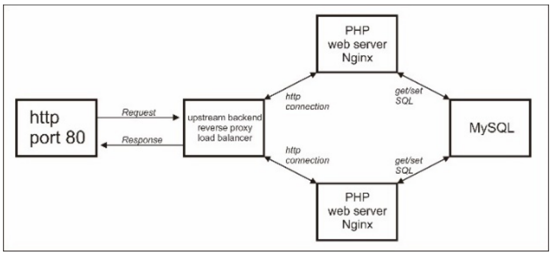

Figure 4. Raspberry Pi Load Balancing Diagram Block

In designing this application, it will be made using programming methods with native PHP language. This is done as an effort to maximize performance on load balancing. The structure of the web application page can be seen in Figure 5.

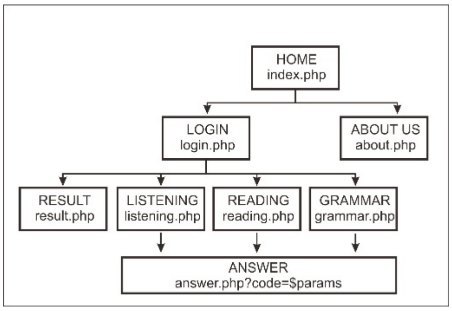

Figure 5. Web Application Page Structure

4.4 Implement

The first implementation step is setting up an Internet Protocol (IP) address. The IP address of each device that will be implemented in the topology can be seen in Table 3.

Talel 3. IP Address Configuration

Device Name	IP Address	Subnet Mask	Gateway
Load Balancer	Eth0: 192.168.10.1 Eth1: 192.168.0.1	255.255.255.240 255.255.255.0	- -
Web Server 1	Eth0: 192.168.10.2	255.255.255.240	192.168.10.1
Web Server 2	Eth0: 192.168.10.3	255.255.255.240	192.168.10.1
Server Database	Eth0: 192.168.10.4	255.255.255.240	192.168.10.1
Computer Admin	Eth0: 192.168.10.5	255.255.255.240	192.168.10.1
User's Computer	Eth0: 192.168.0.2 – 192.168.0.40	255.255.255.0	192.168.10.1

IP Address Configuration setting is as follows :

```
pi@raspberrypi:~$ sudo nano
/etc/dhcpcd.conf
```

The following is the IP address configuration on the file "/etc/dhcpcd.conf".

```
interface eth0 static
ip_address=192.168.10.2/29
static routers=192.168.10.1
```

The next step after setting up an IP is to install the Nginx web server. To install the Nginx web server on Raspberry Pi, the following commands were used :

```
pi@raspberrypi:~$ sudo apt-get
install nginx
```

After that, the PHP programming language must be installed manually on the Raspberry Pi. PHP must be installed because the computer-based web application that will be used is a PHP-based programming language application. The way to install PHP on Raspberry Pi is with the following command:

```
pi@raspberrypi:~$ sudo apt-get
install php5 php5-fpm php5-mysql
```

The above command is the command to install php5 as the basis for computer-based exam application processor, php5-fpm is to manage PHP with Nginx web server and php5-mysql is used as a database module in computer-based exam applications.

After the command is finished, the next step is to enable PHP to run on Nginx web server. The commands used are as follows:

```
pi@raspberrypi:~$ sudo nano
/etc/nginx/site-enabled/default
```

The next step is load balancing configuration using Nginx web server. The file that will be changed is the file "/etc/nginx/nginx.conf". The way to change for the configuration is with the following command:

```
pi@raspberrypi:~$ sudo nano
/etc/ngin/nginx.conf
```

Change the file to the following below :

```
http {
  upstream backend {
    server 192.168.10.2;
    server 192.168.10.3;
}
server {
   listen 80;
   location / {
       proxy_pass http://backend;
   }
}
```

The script on the server 192.168.10.2 and server 192.168.10.3 is the IP address of the web server that will be the user's main destination containing computer-based exam application. After adding the script, press Ctrl + X then select "yes" to save the changes. In order for this command to be executed, we must delete the default command by:

```
pi@raspberrypi:~$ sudo rm
/etc/nginx/site-enable/default
```

Don’t forget to re-start the web server, so that the changed script can run.

4.5 Operate

At this stage, several scripts will be created on the connected connection to see which web server is actively serving the user. The script is using PHP as follows:

```
<?php
header('Content-Type: text/plain');
session_start();
echo “Web Server 1 <br>”;
echo "IP Server:
".$_SERVER['SERVER_ADDR'];
    ?>
```

4.6 Optimize

To find out the performance optimization of load balancer and web server, it will be tested with Web Server Stress Tool software. This can generate reports in the form of data from each user connection.
The following are the scenarios that will be carried out to optimize performance optimization:

a. Scenario 1
Method : *Load Balancing*
Device : Raspberry PI 3 Model B
Optimization Duration : 60 minutes
Number of Simulation Users : 50 *users*
Each User’s Click Time : 5 seconds

b. Scenario 2
Method : *Load Balancing*
Device : Raspberry PI Model B
Optimization Duration : 60 minutes
Number of Simulation Users : 100 *users*
Each User’s Click Time : 5 seconds

c. Scenario 3
Method : *Load Balancing*
Device : Raspberry PI 3 Model B
Optimization Duration : 60 minutes
Number of Simulation Users : 150 *users*
Each User’s Click Time : 5 seconds

d. Scenario 4
Method : *Load Balancing*
Device : Raspberry PI Model B
Optimization Duration : 60 minutes
Number of Simulation Users : 200 *users*
Each User’s Click Time : 5 seconds

e. Scenario 5
Method : *Single Server*
Device : MSI i5-6400 CPU @ 2.70GHz 16GB
Optimization Duration : 60 minutes
Number of Simulation Users : 200 *users*
Each User’s Click Time : 5 seconds

4 RESULT AND DISCUSSION

The following are the test results and the results of the implementation and discussion that has been carried out. This discussion will be explained based on each research factors.

1. Test Result for Each User
   This test is intended to see the average of each research scenario, which can be seen in Table 4.

Table 4. Average Tsest on each user

Scenario	***Clicks***	***Hits***	***Errors***	***Avg. Click Time [ms]***
I	616,78	616,02	0	757,06
II	548,54	547,56	0,02	1464,87
III	489,72	488,8	0	2245,29
IV	461,73	461,41	0,01	2711,67
V	562,46	561,57	0	1284,99

From these data it can be concluded that by using load balancing method, the more the number of users, it will affect the average response time for each click. The more number of users, the longer the response time for each click.

In scenario 5 using a single server and not using Raspberry PI as a server (in contrast to the previous 4 scenarios) on 200 users has almost the same value as scenario 2 that uses load balancing with 100 users. Whereas when compared with scenario 4, also with 200 users, scenario 5 looks better with 1426.68 ms faster in the average response time for each click.

2. Test Result for each page

The list of pages to be tested is in table 5.

Table 5. Test page

No	**URL**	**POST data (or @filename@)**
1	http://192.168.10.1/	
2	http://192.168.10.1/ login.php	
3	http://192.168.10.1/ login.php	username=demo& password =demo&submit=Sign+in
4	http://192.168.10.1/ index.php	
5	http://192.168.10.1/ listening.php	
6	http://192.168.10.1/ reading.php	
7	http://192.168.10.1/ grammar.php	
8	http://192.168.10.1/ result.php	
9	http://192.168.10.1/ about.php	
10	http://192.168.10.1/ answers.php? code=Listening	
11	http://192.168.10.1/ answers.php? code=Reading	
12	http://192.168.10.1/ answers.php? code=Grammar	

The following are the test results of each page.

a. Scenario 1

Table 6. Scenario 1 website page testing

URL No.	***Clicks***	***Errors***	***Errors [%]***	***Time Spent [ms]***	***Avg. Click Time [ms]***
1	3.187	0	0	1.711.964	537
2	3.192	0	0	1.684.130	528
3	2.239	0	0	2.923.020	1.306
4	2.553	0	0	1.354.100	530
5	2.435	0	0	1.947.686	800
6	2.429	0	0	1.969.809	811
7	2.433	0	0	1.951.161	802
8	2.387	0	0	2.181.128	914
9	2.556	0	0	1.336.245	523
10	2.429	0	0	1.970.465	811
11	2.430	0	0	1.969.450	810
12	2.421	0	0	2.007.114	829
AVG	**2.558**	**0**	**0**	**1.917.189**	**767**

b. Scenario 2

Table 1. Scenario 2 website page testing

URL No.	**Clicks**	**Errors**	**Errors [%]**	**Time Spent [ms]**	**Avg. Click Time [ms]**
1	4.923	4.714	95,75	3.372.342	16.136
2	4.948	4.730	95,59	3.117.262	14.299
3	4.899	4.758	97,12	210.988	1.496
4	4.955	4.741	95,68	3.037.322	14.193
5	4.358	4.199	96,35	2.324.231	14.618
6	4.340	4.182	96,36	2.088.325	13.217
7	4.340	4.181	96,34	2.102.674	13.224
8	4.374	4.219	96,46	1.949.025	12.574
9	4.411	4.218	95,62	2.766.360	14.333
10	4.370	4.212	96,38	2.027.006	12.829
11	4.335	4.179	96,4	1.968.268	12.617
12	4.346	4.190	96,41	1.901.862	12.191
AVG	**4.550**	**4.377**	**96**	**2.238.805**	**12.644**

c. Scenario 3

Table 2. Scenario 3 website page testing

URL No.	***Clicks***	***Errors***	***Errors [%]***	***Time Spent [ms]***	***Avg. Click Time [ms]***
1	6.941	32	0,46	10.442.384	1.511
2	6.935	34	0,49	10.499.925	1.522
3	5.262	43	0,82	19.119.125	3.663
4	6.938	29	0,42	10.436.747	1.511
5	6.096	27	0,44	14.952.281	2.464
6	6.098	31	0,51	14.941.108	2.463
7	5.637	32	0,57	13.725.211	2.449
8	5.546	33	0,6	14.135.478	2.564
9	6.403	30	0,47	9.652.627	1.515
10	5.630	32	0,57	13.747.329	2.456
11	5.614	27	0,48	13.858.413	2.480
12	5.622	28	0,5	13.827.995	2.472
AVG	**6.060**	**32**	**1**	**13.278.219**	**2.256**

d. Scenario 4

Table 3. Scenario 4 website page testing

URL No.	***Clicks***	***Errors***	***Errors [%]***	***Time Spent [ms]***	***Avg. Click Time [ms]***
1	8.717	549	6,3	14.439.270	1.768
2	8.700	526	6,05	14.573.025	1.783
3	6.450	460	7,13	26.284.810	4.388
4	8.713	566	6,5	14.393.945	1.767

5	7.464	408	5,47	21.190.571	3.003
6	7.422	375	5,05	21.479.664	3.048
7	7.481	427	5,71	21.133.943	2.996
8	7.348	363	4,94	21.827.892	3.125
9	8.173	488	5,97	13.799.477	1.796
10	7.029	401	5,7	19.926.871	3.006
11	7.025	376	5,35	20.043.314	3.014
12	6.989	360	5,15	20.165.396	3.042
AVG	**7.626**	**442**	**6**	**19.104.848**	**2.728**

e. Scenario 5

Table 4. Scenario 5 website page testing

URL No.	***Clicks***	***Errors***	***Errors [%]***	***Time Spent [ms]***	***Avg. Click Time [ms]***
1	9.530	0	0	12.079.092	1.267
2	10.276	0	0	8.284.757	806
3	8.323	0	0	18.310.717	2.200
4	10.284	0	0	8.229.207	800
5	9.408	0	0	12.715.358	1.352
6	9.410	0	0	12.696.090	1.349
7	9.414	0	0	12.705.133	1.350
8	9.362	0	0	13.037.187	1.393
9	9.670	0	0	7.792.409	806
10	8.851	0	0	12.004.091	1.356
11	8.847	0	0	12.045.974	1.362
12	8.850	0	0	12.052.230	1.362
AVG	**9.352**	**0**	**0**	**11.829.354**	**1.284**

In the tables above, the number of clicks is different. Average errors, time spent on requests, and average time for each click.

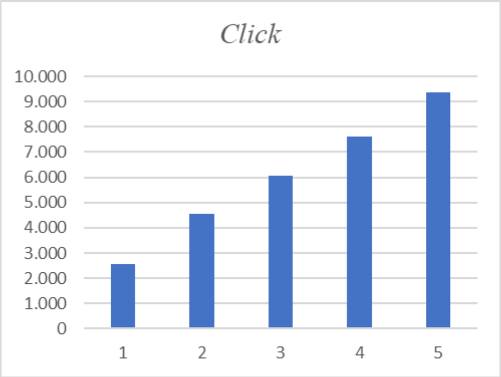

Figure 1. Testing on the amount of clicks

The biggest amount of clicks is occurred in scenario 4 as many as 7,626 and scenario 1 gets the lowest number of clicks which is only 2,558. This can happen because the more number of users, the more the number of clicks will be.

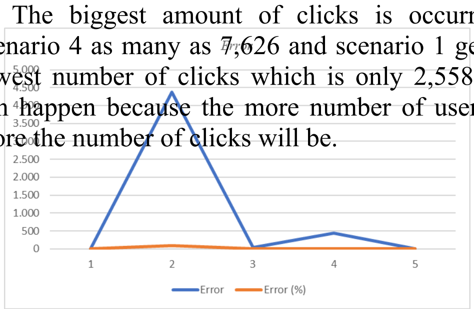

Figure 2. Testing on the amount of errors

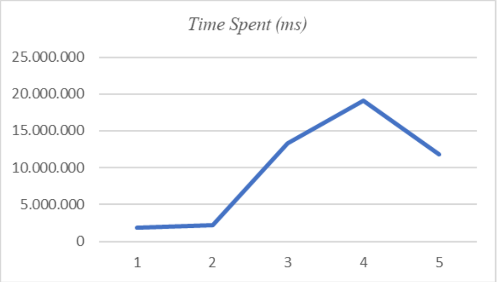

Figure 3. Testing on the amount of Time Spent (ms)

The time spent on the website for each user request follows the number of users themselves. The more the users, the busier the website's response will be. This is illustrated by the table where the total time spent in scenario 4 is greater because in this scenario the number of users reaches 200 within 60 minutes. Greater than other scenarios.

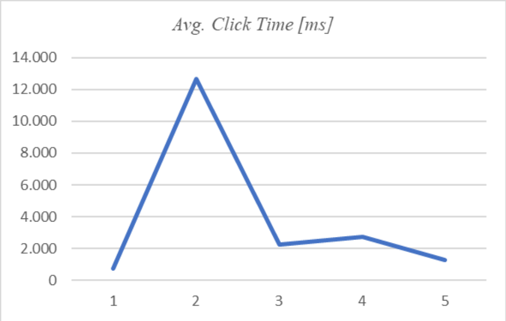

Figure 4. Testing on the amount of Average Click Time (ms)

The average time for each click in scenario 2 is the biggest, which is 12,644 ms, this is because in scenario 2 there is an error during testing.

3. Server and Bandwidth Test Result

a. Scenario 1

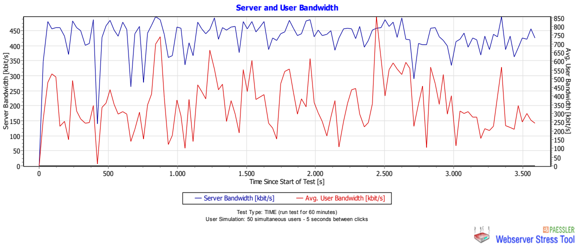

Figure 5. Scenario 1 server and user bandwidth test

The average bandwidth is so large, and in scenario 1 it can be seen that the bandwidth of the server is much larger.

b. Scenario 2

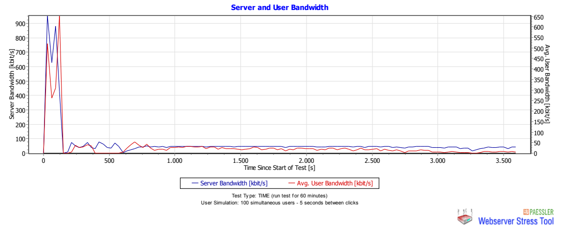

Figure 6. Scenario 2 server and user bandwidth test

Seen in scenario 2 the amount of bandwidth wasn't stable, this is affected because in scenario 2 there are many up to 96%.

c. Scenario 3

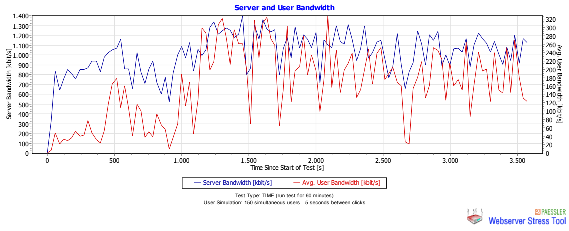

Figure 7. Scenario 3 server and user bandwidth test

In Scenario 3, there were an increase in user's average bandwidth

.

d. Scenario 4

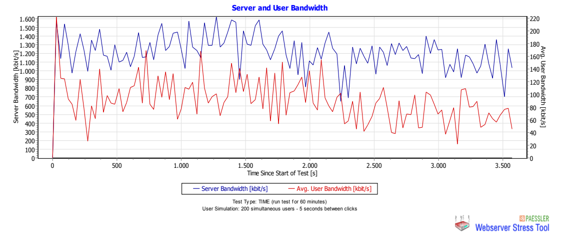

Figure 8. Scenario 4 server and user bandwidth test

e. Scenario 5

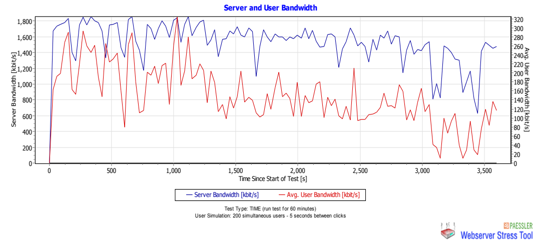

Figure 9. Scenario 5 server and user bandwidth test

From some of these results, it can be concluded that the bandwidth of the clustering load balancing system can meet bandwidth with up to 200 users in 60 minutes.

4. Results of Data Transfer Test, Memory System, and CPU Load

In the graphs below you will see traffic on the network that is related to memory and the load that will be received by the server CPU. The following is the graph :

a. Scenario 1

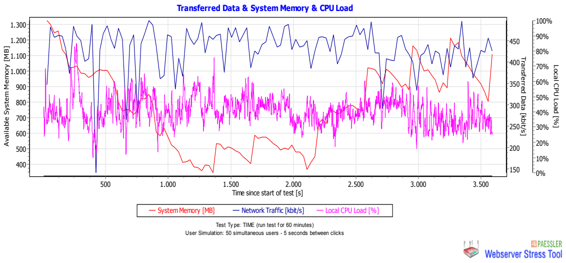

Figure 10. Scenario 1data transfer, memory system, and CPU load test

b. Skenario 2

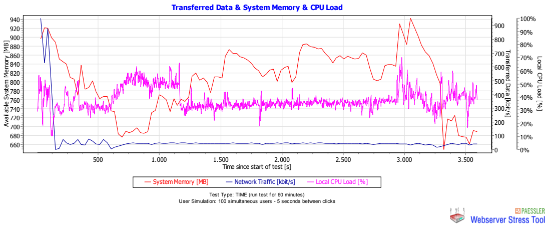

Figure 11. Scenario 2 data transfer, memory system, and CPU load test

c. Skenario 3

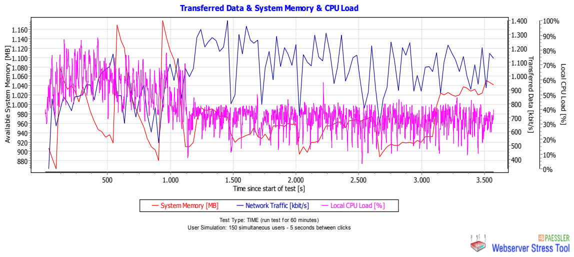

Figure 12. Scenario 3 data transfer, memory system, and CPU load test

a. Scenario 4

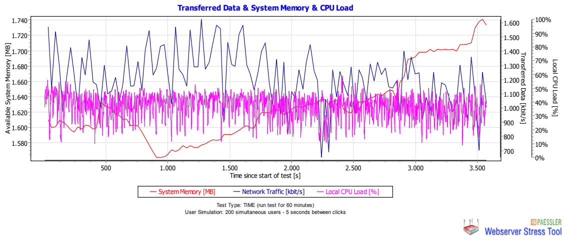

Figure 13. Scenario 4 data transfer, memory system, and CPU load test

b. Scenario 5

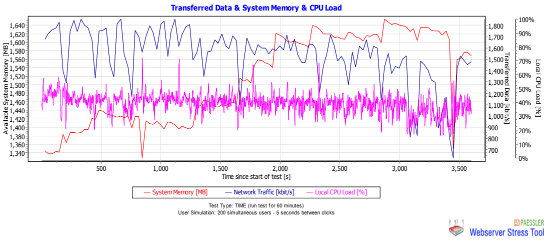

Figure 14. Scenario 5 data transfer, memory system, and CPU load test

4 CONCLUSIONS

The following are the conclusions obtained from the results of this study. The conclusions are as follows:

1. 1. The use of Raspberry Pi can be implemented in the computer lab of SMPN 89 Jakarta with 30 students per class.
2. 2. Raspberry Pi with Raspbian / Linux system operation can be used as a load balancer, web server, database server, FTP server and as a cluster system
3. 3. The higher the number of users will affect the average response time for each click and the response time for each click will be longer
4. 4. The number of clicks will be directly proportional to the number of users. Where if there are many users, then the click received by the server is even greater.
5. 5. The higher the number of users, the website's response will be busier
6. The percentage of errors on the server will affect the average click time. The more errors on the server, the greater the number of average click time that the user receives.
7. Stable bandwidth in each test indicates that the bandwidth in the clustering system with one Raspberry Pi as load balancer, two Raspberry Pi as web servers and one Raspberry as database server is enough to meet the number of users as many as 200 users for 60 minutes.
8. Traffic on load balancing is still stable with a period of 60 minutes and the number of users is between 1 - 200.

REFERENCES

Aivaliotis, D., 2013. *Mastering NGINX.* Birmingham: Packt Publishing Ltd.

Alsyaibani, O. M. A., 2013. Performa Algoritma Load Balance Pada Server Web Apache dan Nginx Dengan Database PostgresSQL. *Skripsi,* p. 26.

Anthony, B. & Jordan, S., 2011. *CCDA 640-864 Official Cert Guide.* Indianapolis: Cisco Press.

Apriyatna, G. A., 2015. *Aplikasi Computer Server dan Clustering Dengan Simulasi Virtual Box Pada PDAM Kota Surabaya,* Surabaya: STIKOM Surabaya.

B. M. Moniruzzaman, M. Waliullah & M. S. Rahman, 2015. A High Availability Clusters Model Combined with Load Balancing and Shared Storage Technologies for Web Servers. *International Journal of Grid Distribution Computing.*

Chandio, A. A. et al., 2014. A comparative study on resource allocation and energy efficient job scheduling strategies in large-scale parallel computing systems. *Springer US Cluster Computing Journal,* XVII(4).

Connoly, T. M. & Begg, C. E., 2010. *Database System a Practical Approach to Design, Implementation and Management.* Boston: Pearson Educato.

Djanali, S. & Studiawan, H., 2013. Power Optimization in. *The 7th International Conference on.*

Gandhi, G. & Garg, S., 2014. *Implementing Software Testing Model Approach for Efficient Bug Finding with Yin-Yang Testing Theory on Java Application.* India: Department of Computer Science RIMT-MAEC.

Guizhi, X., 2015. The Design of a Client-Cloud Computing. *International Journal of Grid and Distributed,* VIII(5).

Kahanwal, B. & Singh, T. P., 2012. The Distributed Computing Paradigms: P2P, Grid, Cluster, Cloud, and Jungle. *International Journal of Latest Research in Science and Technology,* I(2), pp. 183-184.

Kurniawan, H. & Widiyanto, E. P., 2016. Analisis Peningkatan Performa Akses Website dengan Web Server Stress Tool. *Jatisi,* Volume II.

Lukitasari, D. & Fali, O. A., 2010. Analisis Perbandingan Load. *Jurnal Generic,* 5(2).

O'Brien, James, A. & Marakas, G. M., 2011. *Sistem Informasi Manajemen.* Jakarta: Salemba Empat.

Paessler, A., 2014. Webserver Stress Tool Manual.

Pribadi, P. T., 2013. Implementasi High-Availability VPN Client Pada Jaringan Komputer Fakultas Hukum Universitas Udayana. *Jurnal Ilmu Komputer,* VI(1).

Probowo, T., 2014. *Penerapan Intrusion Detection System Pada Web Server Menggunakan Metode Signature Based.* Bandung: Universitas Komputer Indonesia.

Putra, R. H. & Sugeng, W., 2016. Implementasi Cluster Server pada Raspberry Pi dengan Menggunakan Metode Load Balancing. *Jurnal Edukasi dan Penelitian Informatika (JEPIN),* Juni.II(1).

Rao, G. P., Brueggemann, E. R. & Rodriguez, R. A., 2010. *Method for Maintaining Transaction Integrity Across Multiple Remote Access Servers.* US, Paten No. 11/626,334.

Sean, W., 2011. *Designing for Cisco Internetwork Solutions (DESGN) Foundation Learning Guide: (CCDA DESGN 640-864).* 3rd ed. Indianapolis: Cisco Press.

Silberschatz, A., Galvin, P. & Gagne, G., 2010. *Operating System Concepts.* United States of America: John Wiley & Sons.

Solihin, A., 2010. *MySQL 5 Dari Pemula Hingga Mahir.* Jakarta: Achmadtim.ne.

Sosinsky, B., 2009. *Networking Bible.* st ed. Indianapolis: John Wiley & Sons.

Susanti, A., 2016. *Penyebab UN Berbasis Komputer Disenangi Siswa.* [Online]
Available at: https://news.okezone.com/read/2016/04/05/65/1354829/penyebab-un-berbasis-komputer-disenangi-siswa

Vermaat, S. C., 2011. *Discovering Computers "Menjelajah Dunia Komputer" Fundamental.* 3rd ed. Jakarta: Salemba Infotek.

Zhou, Z., Lifa, W. & Zheng, H., 2013. Context-Aware Access. *International Journal of Grid and,* VI(6).